

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-61. (Canceled)

62. (Currently Amended) A method by which a disk-based distributed data storage system attached to a network is organized for protecting historical records of stored data entities and enforcing rules governing the retention of these records, the method comprising:

storing and retrieving data entities, in response to requests by programs that are clients of the distributed data storage system;

recording distinct states of stored data entities, corresponding to different moments of time, as a plurality of entity versions coexisting within the distributed data storage system;

storing portions of an entity version that is one of the plurality of entity versions at each of a plurality of storage sites of the distributed data storage system, in response to a request by a program that is a client of the distributed data storage system to deposit the entity version;

sharing among the plurality of storage sites a set of rules that restrict deletion of the entity version, the sharing comprising:

communicating the set of rules over the network to the plurality of storage sites;

and

storing information derived from the shared set of rules at each of the plurality of storage sites; and

applying the shared set of rules independently at each of the plurality of storage sites, in response to a request by the client program, in order to separately determine whether or not the portion of the entity version at each site can be deleted;

wherein each of the portions represents at least part of the ~~contents of~~ information contained in the entity version, and not all of the portions are needed to reconstruct the entire contents of the entity version;

wherein a failure event occurs that ~~causes a~~ affects just one of the plurality of storage sites and causes it to delete its portion of the entity version in violation of the shared set of rules, and the entity version is subsequently retrieved successfully in response to a retrieval request sent by the client program to the distributed data storage system;

wherein the client program communicates with the disk-based distributed data storage system only over the network;

wherein a first request sent by the client program communicating with the disk-based distributed data storage system causes the shared set of rules to restrict deletion of the entity version at each of the plurality of storage sites;

wherein a second request, sent by the client program after the first request, would enable the entity version to be deleted from all of the plurality of storage sites in violation of the restriction caused by the first request, and the second request is denied;

wherein no request sent by the client program over the network can enable deletion of the entity version to occur in violation of the restriction caused by the first request;

wherein the shared set of rules restrict deletion, based at least in part upon a time that was associated with the entity version in response to a request by the client program; and

wherein a third request, sent by the client program after the second request, causes the portions of the entity version stored at the plurality of storage sites to be deleted.

63. (Canceled)

64. (Previously Presented) The method of claim 62 in which the time associated with the entity version is an expiration time assigned to the entity version, and the assignment is made independently within each of the plurality of storage sites, according to the shared set of rules, before which time both modification and deletion are prohibited.

65. (Original) The method of claim 62 in which no single individual is given the authority to override the deletion prohibition at all of the plurality of storage sites.

66. (Previously Presented) The method of claim 62 in which applying the set of rules at a one of the plurality of storage sites determines that an entity version can be deleted and a portion of the entity version is deleted immediately and storage space that was used to store the portion becomes available to store new data.

67. (Previously Presented) The method of claim 62 in which applying the set of rules determines that an entity version can be deleted but the entity version is not deleted until deletion is requested by a client of the disk-based distributed data storage system.

68-166. (Canceled)

167. (Previously Presented) The method of claim 62 in which, during a time interval, the shared set of rules prohibits deletion of the entity version while others of the plurality of entity versions are allowed to be deleted.

168. (Previously Presented) The method of claim 167 in which the time interval is at least a year in length.

169. (Canceled)

170. (Previously Presented) The method of claim 167 in which the client program causes the time interval during which deletion is prohibited to be extended and no subsequent action taken by the client program can cause the time interval to be shortened.

171. (Previously Presented) The method of claim 167 in which the client program causes the length of the time interval to be set and no subsequent action taken by the client program can shorten the time interval.

172. (Previously Presented) The method of claim 171 in which the length of the time interval is initially not set and, before the length of the time interval is set, no action taken by the client program can cause the entity version to be deleted

173. (Previously Presented) The method of claim 167 in which no action taken by any client program that only communicates with the disk-based distributed data storage system over the network can cause the time interval to be shortened.

174. (Previously Presented) The method of claim 62 in which the plurality of entity versions record historical states of a single stored data entity, with each of the plurality of entity versions associated with a historical time interval during which the recorded historical state was the state of the single stored data entity.

175. (Previously Presented) The method of claim 174 in which the shared set of rules that determine whether or not the entity version can be deleted depend at least in part on the length of the historical time interval associated with the entity version.

176. (Previously Presented) The method of claim 174 in which the shared set of rules that determine whether or not the entity version can be deleted depend at least in part on whether or not the historical time interval associated with the entity version includes a specified moment of time.

177. (Previously Presented) The method of claim 62 in which a stored data entity is a file in a file system or a record in a database or an object in an object storage system.

178. (Previously Presented) The method of claim 62 in which two of the plurality of storage sites are located in different cities.

179. (Previously Presented) The method of claim 62 in which the set of rules are represented in a rule description data structure, and the information derived from the set of rules that is stored at each of the plurality of storage sites comprises a hash of the contents of the rule description data structure.

180. (Previously Presented) The method of claim 62 in which the entity version is a version of a stored data entity and the first request causes a new version of the stored data entity to be stored.

181. (Previously Presented) The method of claim 62 in which the first request assigns an expiration time to the entity version, before which time deletion is prohibited.

182. (Previously Presented) The method of claim 62 in which the second or third request attempts to delete the entity version or to change the time associated with the entity version.

183. (Previously Presented) The method of claim 62 in which the entity version is a version of a stored data entity and the third request causes a new version of the stored data entity to be stored.

184. (Previously Presented) The method of claim 62 in which the time associated with the entity version is a time when the entity version was created, transmitted or stored; or had some property changed; or a time assigned to the entity version.

185. (Previously Presented) The method of claim 62 in which the shared set of rules are communicated to the plurality of storage sites at the time that the client program communicating

with the disk-based distributed data storage system deposits the entity version into the storage system.

186. (Previously Presented) The method of claim 62 in which care is taken to ensure that operators and administrators of the disk-based distributed data storage system have no special privileges or physical access that would allow them to circumvent or change the shared set of rules at all of the plurality of storage sites.

187. (Previously Presented) The method of claim 62 in which both the independent application of the shared set of rules at each of the plurality of storage sites and the manner in which information that determines the rules is communicated between storage sites are designed to prevent alterations or corruptions of the operation at a one of the plurality of storage sites from allowing the entity version to be deleted or modified at another of the plurality of storage sites in violation of the restriction on the deletion of the entity version.

188. (Previously Presented) The method of claim 62 in which the plurality of storage sites communicate with one another over the network in order to achieve fault tolerance against the loss of storage sites.

189. (Previously Presented) The method of claim 62 in which the shared set of rules do not, without additional information, determine a moment of time after which deletion is allowed.

190. (Previously Presented) The method of claim 62 in which the shared set of rules together with a specification of a starting time do not, without additional information, determine a moment of time after which deletion is allowed.

191. (Previously Presented) The method of claim 62 in which the sharing step further comprises: determining the set of rules to share from a request by the client program.